

Leksički procesori

Zadatak

Zadatak 0.1

U programskom jeziku ISPIT postoje dve vrste iskaza:

promenljiva = izraz

IF promenljiva THEN iskaz ENDIF

Identifikatori promenljivih su jednoslovni. Izrazi se sastoje od promenljivih i celobrojnih pozitivnih konstanti povezanih operatorima + i -. IF iskazi se mogu ugnežđavati; iskaz iza THEN se izvršava ako je vrednost promenljive iza IF različita od nule. Sve leksičke jedinice moraju međusobno biti razdvojene prazninama (razmak ili novi red).

a) Konstruisati deterministički konačni procesor koji služi da prepozna jednu leksičku jedinicu (leksemu) i pretvori je u interni leksički kod. Kodovi su dati u sledećoj tabeli (P – promenljiva, C – konstanta):

Klasni deo	P	C	IF	THEN	ENDIF	=	+
Vrednosni deo	redni broj ulaza u tabeli simbola	vrednost	-	-	-	-	-

b) Koji niz kodova odgovara programu: A = X + 2 IF A THEN IF B THEN C = 3 ENDIF ENDIF?

c) Objasniti šta u ovom slučaju radi transliterator.

Analiza problema

Leksički analizator ima ulogu pripreme ulaza za sintaksno-semantičku analizu i prevođenje. Ulaz u leksički analizator je program u obliku niza znakova. Zadatak leksičkog analizatora je da u nizu znakova identifikuje pojavu logički povezanih grupa znakova (leksičkih jedinica, odnosno leksema) i na izlazu izda za svaku prepoznatu leksemu njoj odgovarajući interni leksički kod. Klasni deo internog koda služi da označi tip leksičke jedinice, a vrednosni deo daje eventualne dodatne informacije.

Algoritam funkcionisanja traženog leksičkog analizatora glasi:

- izdvojiti leksemu omeđenu prazninama;
- ako leksema počinje slovom odredi da li se radi o rezervisanoj reči ili promenljivoj;

- za promenljivu: ubaci promenljivu u tabelu simbola i vrati interni kod (p,broj ulaza);
 - za rezervisanu reč vrati odgovarajući interni kod;
- za specijalne znake vrati odgovarajući interni kod;
 - za konstantu da niz znakova sakupi, pretvori u binarnu vrednost i vrati interni kod (c,binarna vrednost).

Rešenje

b)

Niz internih kodova u formi (klasa, vrednost) koji odgovara datom programu je:

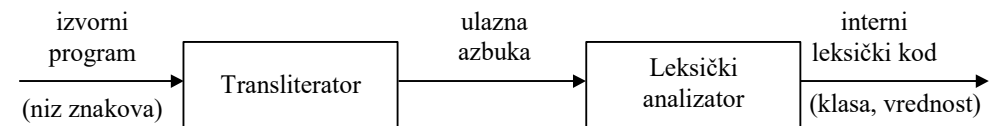
(P,1) (=,-) (P,2) (+,-) (c,2) (IF,-) (P,1) (THEN,-) (IF,-) (P,3) (THEN,-) (P,4) (=,-) (c,3) (ENDIF,-) (ENDIF,-)

U okviru leksičke analize formira se i tabela simbola

broj ulaza	simbol
1	A
2	X
3	B
4	C

c)

- leks. analizator konačni automat; neracionalno za svako posebno slovo posebna kolona ako se sva slova tretiraju na isti način
- zato samo jedan ulazni simbol leks. analizatora “slovo”. Pripremu vrši transliterator



Transliteritor se efikasno realizuje u vidu vektora

48	49	...	64	65	66	122
Cifra	Cifra	...	Spec.	Slovo	Slovo	Slovo
ASCII	ASCII	...	ASCII	ASCII	ASCII	ASCII
("0")	("1")	...	("@")	("A")	("B")	("Z")

a)

Proširenje automata u procesor \Rightarrow svakom ulazu tabele prelaza odgovara jedna akcija. Prazni ulazi označavaju greške u leksičkoj analizi.

	slovo	cifra	spec	—
startno stanje A	B1	C1	D	
nastavlja se ključna reč B	B2	B2		akcija1
nastavlja se konstanta C		C2		akcija2
došao specijalni znak D				akcija3

Akcije:

B1: zn:= 1
bafer[zn]:= vrednosni deo leksičkog koda

B2: zn:=zn+1
bafer[zn]:= tekući znak sa ulaza

C1: BINVR:= ASCII(tekući znak sa ulaza) - ASCII("0")

C2: BINVR:= BINVR * 10 + ASCII(tekući znak sa ulaza) - ASCII("0")

D: SPEC:= klasa tekućeg znaka sa ulaza

akcija1:

1. konsultuj prepoznavać ključnih reči
2. ako je prepoznavać vratio error onda
ako je zn=1 onda ubaci bafer[zn]u tabelu simbola u novi ulaz n i vrati interni kod (p,n)
inače error
inače vrati interni kod (ključna reč, —) koji je vratio prepoznavać ključnih reči

akcija2: vrati interni kod (c,binarna vrednost)

akcija3: vrati interni kod (spec,—)

1	I	2	E ERROR
	F		N ERROR
	--		D ERROR
	"IF"		I ERROR
2	T	3	F ERROR
	H		-- ERROR
	E		"ENDIF"
	E		
	N		
	--		
	"THEN"		

Prepoznavać ključnih reči u vidu proširive liste prelaza

Zadatak

Skup reči koje treba prepoznati je ARRAY, ARCTAN, B1, B2, B3, DOG, CAT, DOGFIGHT. Realizovati prepoznavać na bazi savršene heš funkcije.

Analiza problema

- heš funkcija: preslikava reč (niz znakova) u celobrojni indeks
- "savršena" heš funkcija: svakoj od zadatih reči jedinstven indeks
- prepoznavanje reči: date ključne reči se stave u heš tabelu prema svom indeksu (nema kolizija, može biti praznih ulaza u tabeli). Za reč X koju treba prepoznati izračuna se vrednost H heš funkcije. Pročita se ulaz H tabele. Ako je prazan, ili sadrži reč $Y \neq X$, znači da X nije nijedna od ključnih reči; u suprotnom, X se poklapa sa Y.
- Jedna jednostavna heš funkcija za reč *id* dužine n znakova: zbir po modulu veličine heš tabele HASHSIZE kodova svih znakova pomnoženih multiplikativnom konstantom HASHMUL:

Hash(0) := 0

Hash(t) := Hash(t - 1) * HASHMULT + id(t), $t = 1, 2, \dots, n$

Hash := Hash(n) mod HASHSIZE

- konstante HASHSIZE i HASHMULT određuju se probanjem

Rešenje

Za gornji skup reči dobija se da je HASHMULT = 20, HASHSIZE = 9. Potprogram ključna() vraća indikaciju da li ulazna reč pripada ili ne datom skupu reči.

```
#include <string.h>

static char *table[]={ NULL, "B1", "B2", "B3", "ARCTAN", "CAT",
                        "DOG", "ARRAY", "DOGFIGHT" };

int ključna( const char *rec ) {
    unsigned hash = 0; /* ne valja int */
    const char *tek = rec;
    while ( *tek ) hash = hash * 20 + *tek++;
    return !strcmp( table[ hash % 9 ], rec );
}
```